Fig. 1

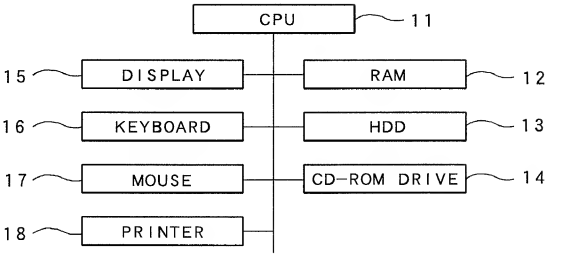


Fig. 2

```
//outline
//   obtain line count of file
// return value
//   line count of file
// explanation of parameter
//   nothing
int  getline()
{
  //initialize variavle
  int  c, nl;
  nl=0;
  // count number of lines till EOF is detected
  while((c=getchar()) !=EOF)
        if(c=='\n')
              ++nl;
  // display line count on screen
  printf("%d\n", nl);
  returun nl;
}
```

Fig. 3

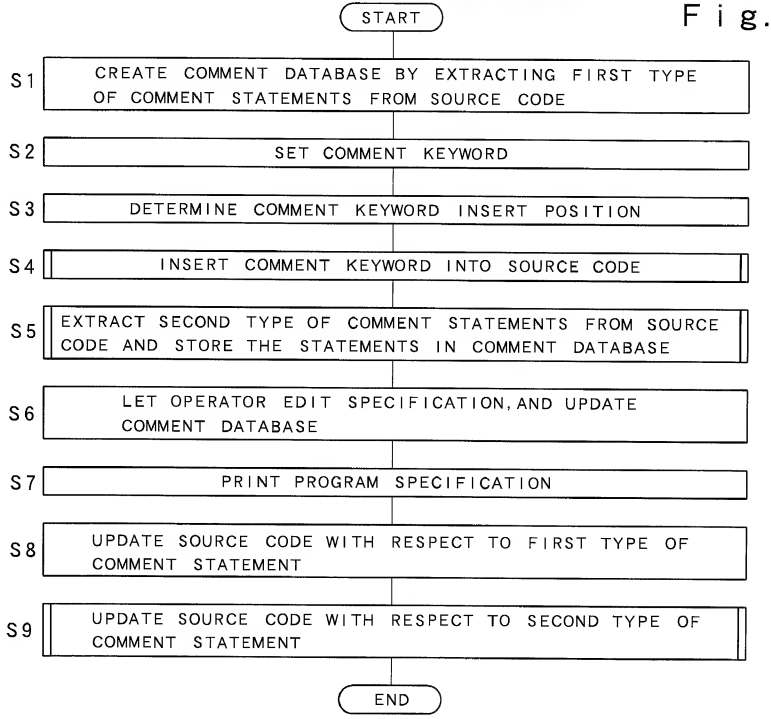


Fig. 4

| IDENTIFIER | COMMENT ITEM | COMMENT |
|---|---|---|
| getline | OUTLINE | obtain line count of file |
| | RETURN VALUE | line count of file |
| | EXPLANATION OF PARAMETER | nothing |
| | FUNCTIONAL EXPLANATION | |

Fig. 5

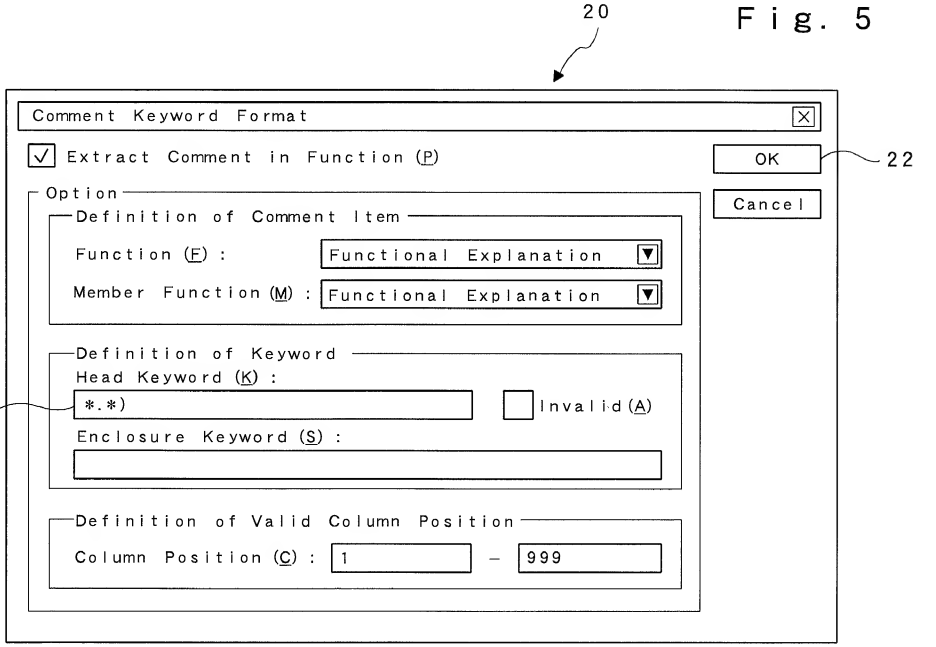


Fig. 6

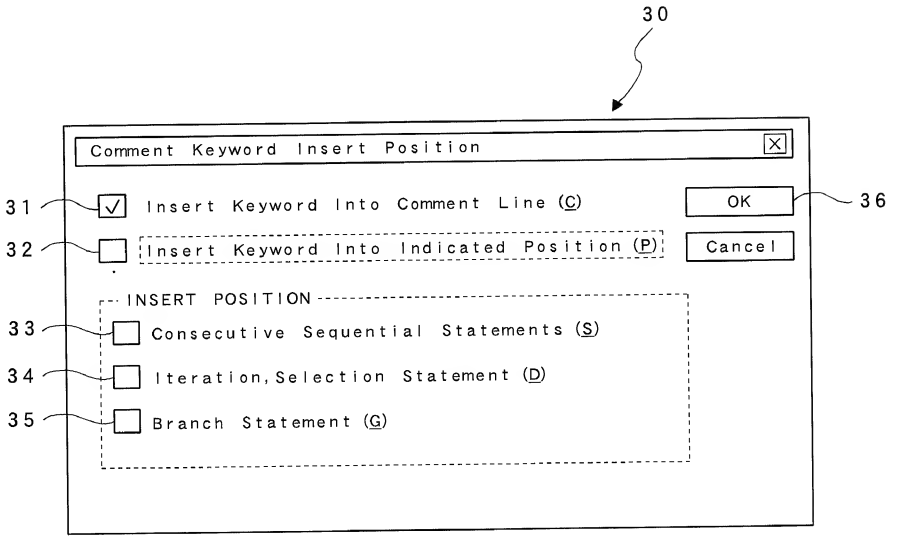

30
Comment Keyword Insert Position
31
Insert Keyword Into Comment Line (C)
32
Insert Keyword Into Indicated Position (P)
OK
36
Cancel
INSERT POSITION
33
Consecutive Sequential Statements (S)
34
Iteration, Selection Statement (D)
35
Branch Statement (G)

Fig. 7

```
//outline
//    obtain line count of file
//  return value
//    line count of file
//  explanation of parameter
//    nothing
int  getline ()
{
   //initialize variavle
   int   c, nl;
   nl=0;
   // count number of lines till EOF is detected
   while ((c=getchar ()) !=EOF)
         // check whether it is line feed signal or not
         if (c=='\n')
                // count number of lines in the case of line feed signal
                ++nl;
   // display line count on screen
   printf ("%d\n", nl) ;
   // return line count to caller
   returun nl;
}
```

Fig. 8

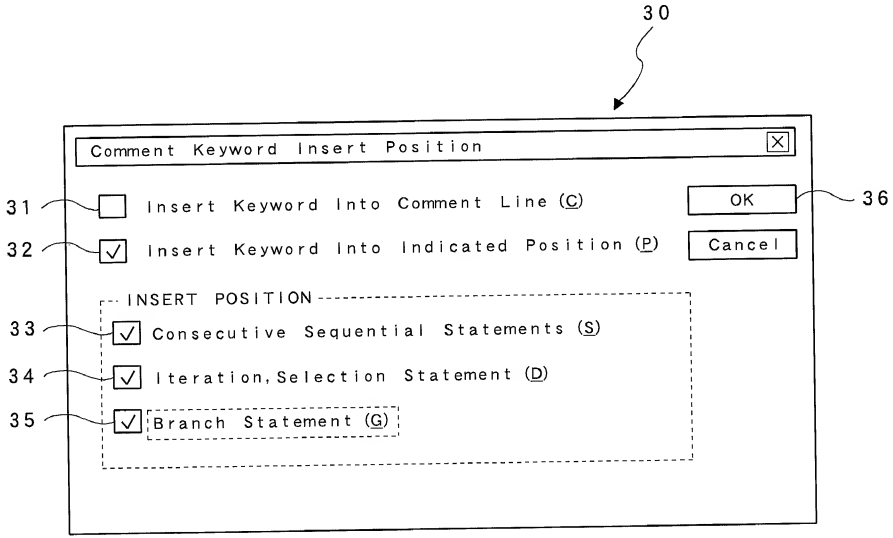

30
Comment Keyword Insert Position
31
Insert Keyword Into Comment Line (C)
32
Insert Keyword Into Indicated Position (P)
OK
Cancel
36
INSERT POSITION
33
Consecutive Sequential Statements (S)
34
Iteration, Selection Statement (D)
35
Branch Statement (G)

Fig. 9

```
//outline
//   obtain line count of file
//  return value
//   line count of file
//  explanation of parameter
//   nothing
int  getline()
{
   int   c, nl;
   nl=0;
   while((c=getchar()) !=EOF)
          if (c=='\n')
                 ++nl;
   printf("%d\n", nl);
   returun nl;
}
```

Fig. 10

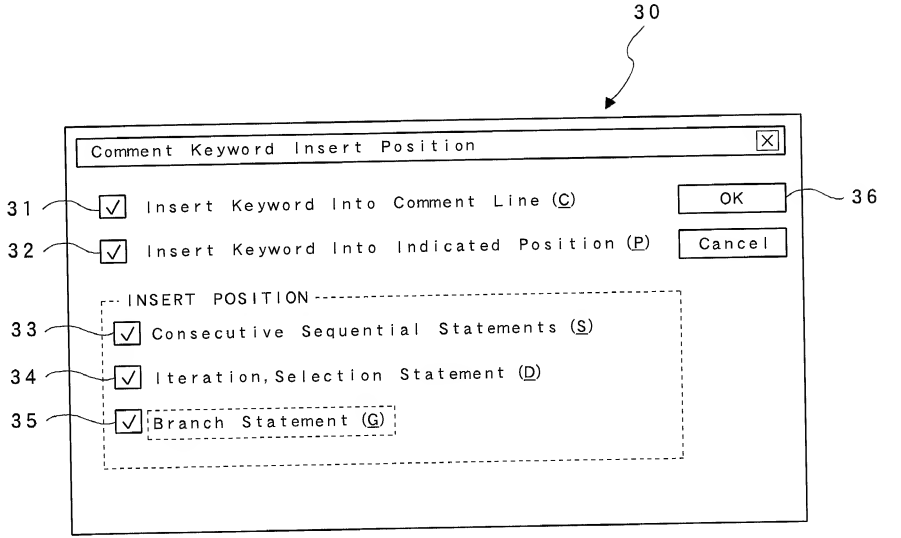

30
Comment Keyword Insert Position
31
Insert Keyword Into Comment Line (C)
32
Insert Keyword Into Indicated Position (P)
INSERT POSITION
33
Consecutive Sequential Statements (S)
34
Iteration, Selection Statement (D)
35
Branch Statement (G)
OK
Cancel
36

Fig. 11

```
//outline
//   obtain line count of file
// return value
//   line count of file
// explanation of parameter
//   nothing
int  getline ()
{
  //1) initialize variavle
  int  c, nl;
  nl=0;
  //2) count number of lines till EOF is detected
  while ((c=getchar ()) !=EOF)
        //2.1) check whether it is line feed signal or not
        if (c=='\n')
              //2.1.1) count number of lines in the case of line feed signal
              ++nl;
  //3) display line count on screen
  printf ("%d\n", nl) ;
  //4) return line count to caller
  returun nl;
}
```

Fig. 12

```
//outline
//   obtain line count of file
// return value
//   line count of file
// explanation of parameter
//   nothing
int  getline()
{
  //1)
  int  c, nl;
  nl=0;
  //2)
  while((c=getchar()) !=EOF)
        //2.1)
        if(c=='\n')
              //2.1.1)
              ++nl;
  //3)
  printf("%d\n", nl);
  //4)
  returun nl;
}
```

Fig. 13

```
//outline
//   obtain line count of file
//  return value
//   line count of file
// explanation of parameter
//   nothing
int  getline()
{
  //1) initialize variavle
  int  c, nl;
  nl=0;
  //2) count number of lines till EOF is detected
  while((c=getchar()) !=EOF)
        //2-1)
        if(c=='\n')
              //2-1-1)
              ++nl;
  //3) display line count on screen
  printf("%d\n", nl);
  //4)
  returun nl;
}
```

Fig. 14

| IDENTIFIER | COMMENT ITEM | COMMENT |
|---|---|---|
| getline | OUTLINE | obtain line count of file |
| | RETURN VALUE | line count of file |
| | EXPLANATION OF PARAMETER | nothing |
| | FUNCTIONAL EXPLANATION | 1) initialize variable<br>2) count number of lines till EOF is detected<br>2. 1)<br>2. 1. 1)<br>3) display line count on screen<br>4) |

Fig. 15

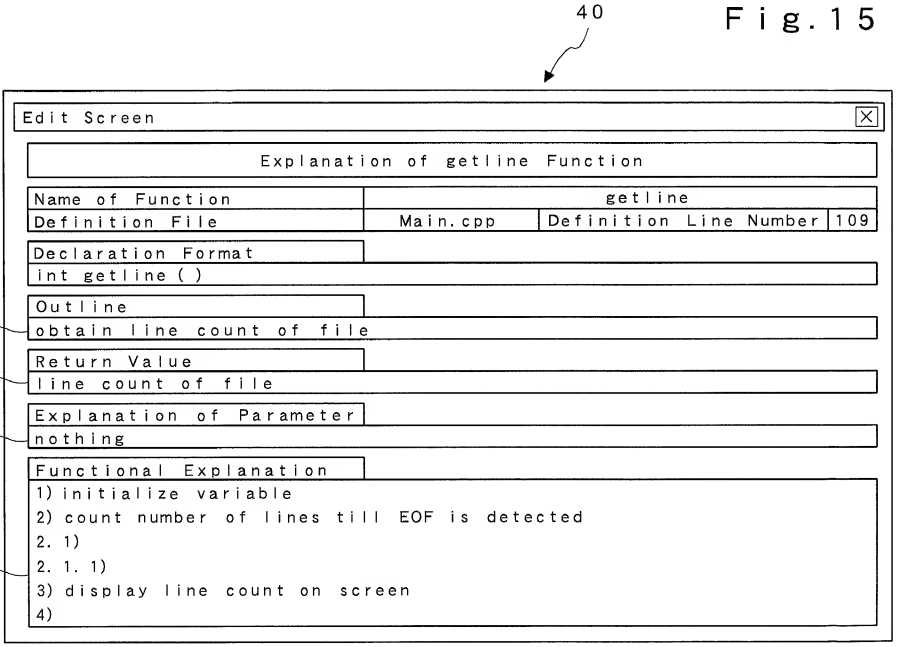


Fig. 16

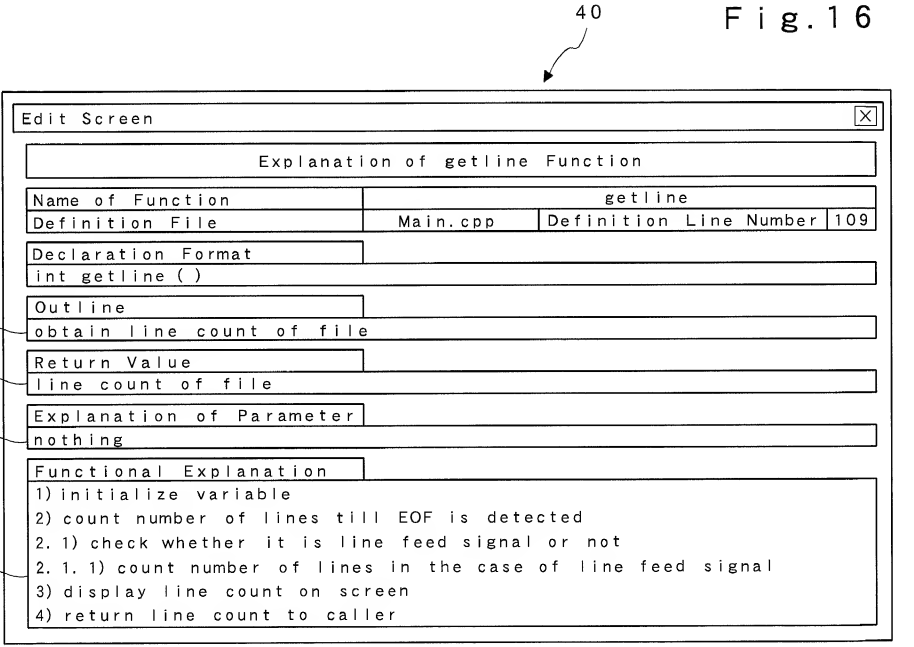


Fig. 17

| IDENTIFIER | COMMENT ITEM | COMMENT |
|---|---|---|
| getline | OUTLINE | obtain line count of file |
| | RETURN VALUE | line count of file |
| | EXPLANATION OF PARAMETER | nothing |
| | FUNCTIONAL EXPLANATION | 1) initialize variable<br>2) count number of lines till EOF is detected<br>2. 1) check whether it is line feed signal or not<br>2. 1. 1) count number of lines in the case of line feed signal<br>3) display line count on screen<br>4) return line count to caller |

Fig. 18

```
//outline
//   obtain line count of file
// return value
//   line count of file
// explanation of parameter
//   nothing
int  getline()
{
  //1) initialize variavle
   int  c, nl;
   nl=0;
   //2) count number of lines till EOF is detected
   while((c=getchar()) !=EOF)
         //2.1) check whether it is line feed signal or not
         if(c=='\n')
                  //2.1.1) count number of lines in the case of line feed signal
                  ++nl;
   //3) display line count on screen
   printf("%d\n", nl);
   //4) return line count to caller
   returun nl;
}
```

Fig. 19

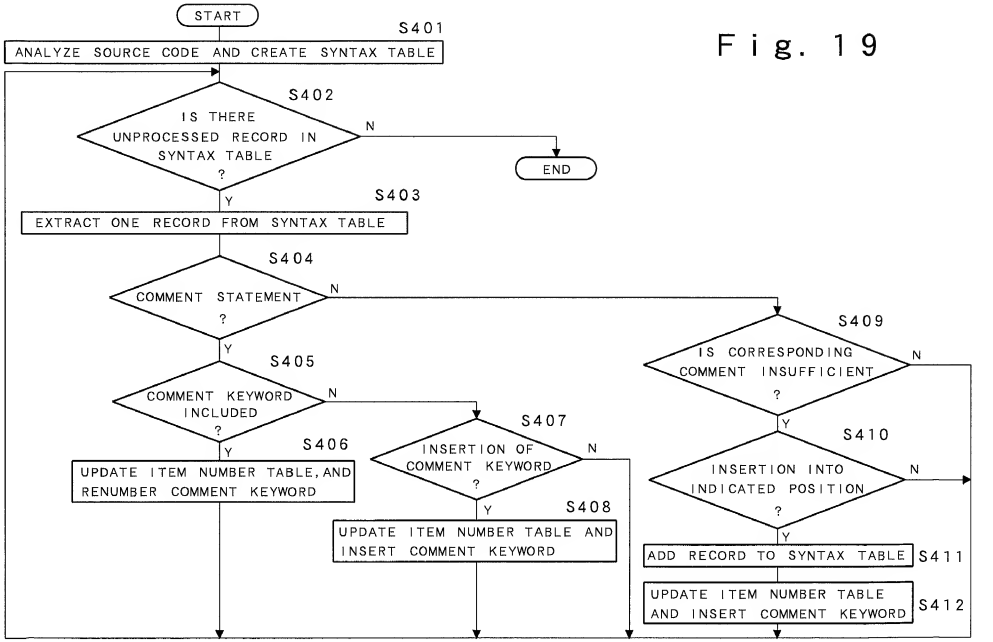


Fig. 20

| STATEMENT CATEGORY | NEST NUMBER | STATEMENT |
|---|---|---|
| COMMENT STATEMENT | 1 | // )initialize variable |
| SEQUENTIAL STATEMENT | 1 | int c, nl; |
| SEQUENTIAL STATEMENT | 1 | nl=0; |
| COMMENT STATEMENT | 1 | // )count number of lines till EOF is detected |
| ITERATION STATEMENT | 2 | while((c=getchar()) !=EOF) |
| SELECTION STATEMENT | 3 | if (c=='\n') |
| SEQUENTIAL STATEMENT | 3 | ++nl; |
| COMMENT STATEMENT | 1 | // ) display line count on screen |
| SEQUENTIAL STATEMENT | 1 | printf("%d\n",nl); |
| BRANCH STATEMENT | 1 | return nl; |

Fig. 21

| STATEMENT CATEGORY | NEST NUMBER | STATEMENT |
|---|---|---|
| COMMENT STATEMENT | 1 | //1) initialize variable |
| SEQUENTIAL STATEMENT | 1 | int c, nl; |
| SEQUENTIAL STATEMENT | 1 | nl=0; |
| COMMENT STATEMENT | 1 | //2) count number of lines till EOF is detected |
| ITERATION STATEMENT | 2 | while ((c=getchar()) !=EOF) |
| COMMENT STATEMENT | 2 | //2.1) |
| SELECTION STATEMENT | 3 | if (c=='\n') |
| COMMENT STATEMENT | 3 | //2.1.1) |
| SEQUENTIAL STATEMENT | 3 | ++nl; |
| COMMENT STATEMENT | 1 | //3) display line count on screen |
| SEQUENTIAL STATEMENT | 1 | printf("%d\n",nl); |
| COMMENT STATEMENT | 1 | //4) |
| BRANCH STATEMENT | 1 | return nl; |

Fig. 22

```
//outline
//   obtain line count of file
//  return value
//   line count of file
//  explanation of parameter
//   nothing
int  getline()
{
  //2)initialize variavle
  int  c, nl;
  nl=0;
  //3) count number of lines till EOF is detected
  while((c=getchar()) !=EOF)
        if (c=='\n')
              ++nl;
  //6)display line count on screen
  printf("%d\n", nl);
  returun nl;
}
```

Fig. 23

| STATEMENT CATEGORY | NEST NUMBER | STATEMENT |
|---|---|---|
| COMMENT STATEMENT | 1 | //2) initialize variable |
| SEQUENTIAL STATEMENT | 1 | int c, nl; |
| SEQUENTIAL STATEMENT | 1 | nl=0; |
| COMMENT STATEMENT | 1 | //3) count number of lines till EOF is detected |
| ITERATION STATEMENT | 2 | while((c=getchar()) !=EOF) |
| SELECTION STATEMENT | 3 | if(c=='\n') |
| SEQUENTIAL STATEMENT | 3 | ++nl; |
| COMMENT STATEMENT | 1 | //6) display line count on screen |
| SEQUENTIAL STATEMENT | 1 | printf("%d\n",nl); |
| BRANCH STATEMENT | 1 | return nl; |

Fig. 24

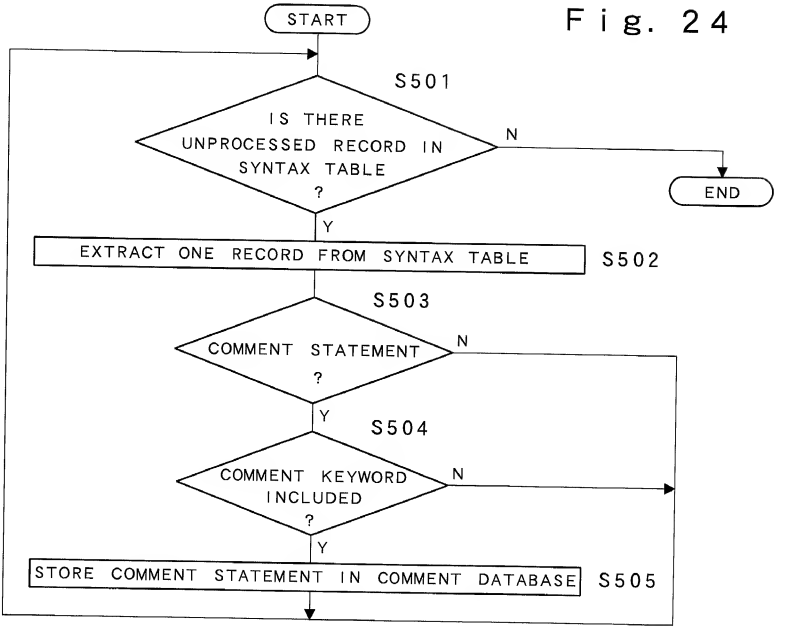
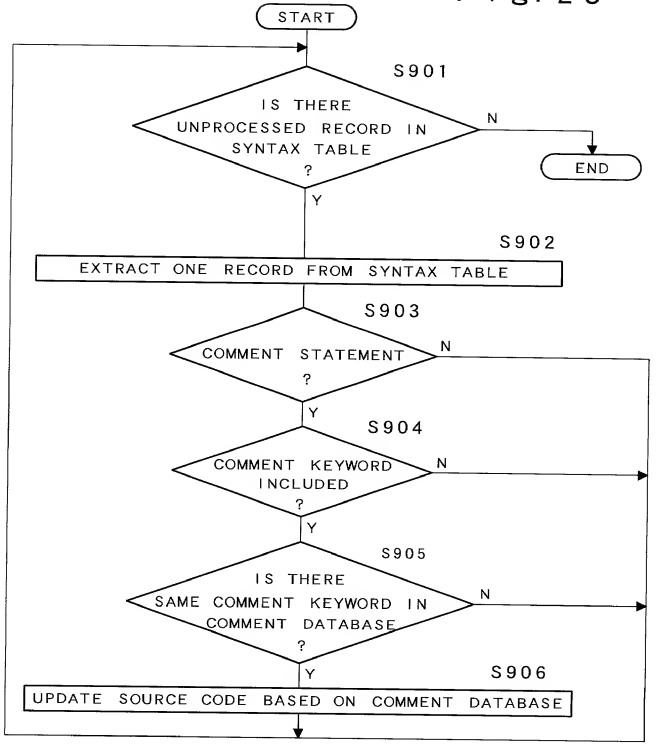


Fig. 25


START
S901
IS THERE UNPROCESSED RECORD IN SYNTAX TABLE ?
N
END
Y
S902
EXTRACT ONE RECORD FROM SYNTAX TABLE
S903
COMMENT STATEMENT ?
N
Y
S904
COMMENT KEYWORD INCLUDED ?
N
Y
S905
IS THERE SAME COMMENT KEYWORD IN COMMENT DATABASE ?
N
Y
S906
UPDATE SOURCE CODE BASED ON COMMENT DATABASE